

```

set "R" (100, 1) // Радиус объекта
//Нечеткие множества параметров
set "все слева" (-0.1, 0;1.75, 0.5;3, 1;3.3, 0)
set "более слева" (-0.1, 0;0.8, 1;1.2, 0)
set "чуть слева" (-0.01, 0;0, 1;0.03, 1;1, 0)
//Нечеткие множества действий
set "чуть повернуть направо" (-0.02, 1)
set "резко повернуть направо" (-0.2, 1)
set "чуть повернуть налево" (0.01, 1)
set "резко повернуть налево" (0.2, 1)

//Правила
if "угол на цель is not "впереди" {
    "скорость корпуса" ~ "стоп"
    if "угол на цель is "все справа" { "поворот корпуса" ~ "резко повернуть направо" }
    if "угол на цель is "все слева" { "поворот корпуса" ~ "резко повернуть налево" }
}

```

Данный формальный язык реализует 2 блока (см. рис. 2): приведение к нечеткости и нечеткий логический вывод с базой правил. 3 блок уже реализуется непосредственно с помощью языков программирования получением данных и данного языка (см. рис. 2).

Описание формального языка будет иметь вид (в форме Бэкуса-Наура)[4, 5]:

```

<prog> ::= <def> <block>
<def> ::= set <term> (fp,fp; fp, fp; fp, fp) | <def> set <term> (fp,fp; fp, fp; fp, fp)
<block> ::= <operator> | <prog>
<operator> ::= <ifoperator> | <setoperator>
<ifoperator> ::= if <cond> { <block> } | if <cond> { <block> } else { <block> }
<cond> ::= <term> | <cond> and <cond> | <cond> or <cond> | (<cond>) | not <cond>
<term> ::= name
<setoperator> ::= <lparam> is <term>
<lparam> ::= name

```

Вероятность срабатывания того или иного правила будет высчитываться по тем же формулам, о которых говорилось ранее.

ЛИТЕРАТУРА

1. Шеври, Ф. Нечеткая логика / Ф. Шеври, Ф. Гели. – Вып. 31.
2. ЭР – Нечеткая логика [Электронный ресурс]. – Режим доступа: http://fuzzy-group.narod.ru/files/Fuzzy_Modeling/Lecture07.Fuzzy.logic.pdf. – Дата доступа: 20.08.2015.
3. ЭР – Математические основы нечеткой логики [Электронный ресурс]. – Режим доступа: <http://bourabai.ru/tpoi/fuzzy.htm>. – Дата доступа: 20.08.2015.
4. ЭР – Формальный язык [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA. – Дата доступа: 20.08.2015.
5. ЭР – форма Бэкуса-Наура [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0_%D0%91%D1%8D%D0%BA%D1%83%D1%81%D0%B0_%E2%80%94%D0%9D%D0%B0%D1%83%D1%80%D0%B0. – Дата доступа: 20.08.2015.

УДК 004

ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ АГЕНТА ПРИ ОБХОДЕ ПРЕПЯТСТВИЙ С ИСПОЛЬЗОВАНИЕМ НЕЧЕТКОЙ ЛОГИКИ

В.А. ПЛЯСОВ

(Представлено: канд. техн. наук, доц. Д.О. ГЛУХОВ)

Рассматриваются проблемы при обходе препятствий агентами с использованием нечеткой логики и дальнейшие способы их решения, которые максимально приближают поведение агента к идеальным случаям.

При использовании данной логики при обходе препятствий возникают следующие проблемы:

1. Реализация плавного обхода препятствия/й.
2. Реализация обхода 2-ух впереди стоящих препятствий.
3. Реализация обхода вдоль границы карты в игре.

Непосредственно рассмотрим каждую проблему в отдельности.

На рисунке 1 показано как будет вести себя агент (юнит) пользуясь обычными нечеткими правилами при обходе одного препятствия.

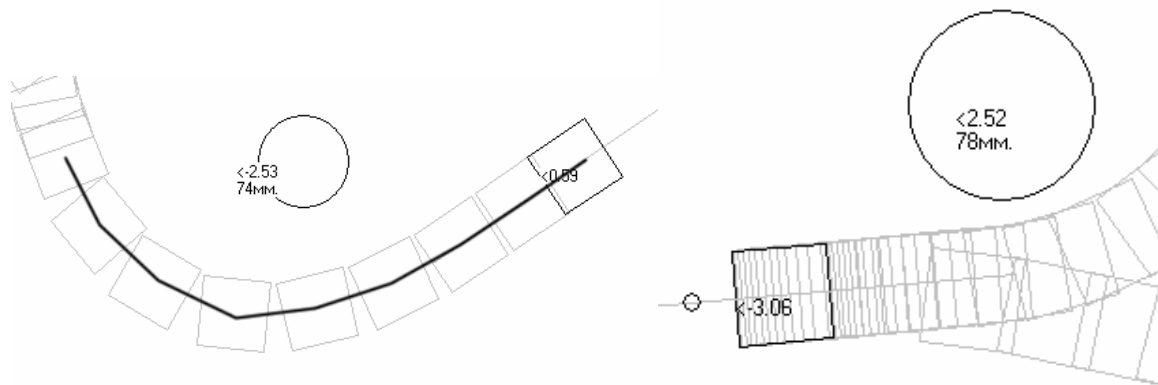


Рис. 1. Плавный обход препятствия агентом

Как говорилось ранее у каждого условия, есть вероятность срабатывания. При любой допустимости срабатывания, даже пусть и при 0.1 сработает какое-либо условие, то в траектории движения будут появляться изгибы. Для того чтобы решить данную проблему и подойти к плавному обходу, необходимо установить порог на вероятность срабатывания, т.е. допустим, установим данный порог на 0.2 из 1, если вероятность срабатывания какого-либо условия меньше данного порога, то оно никогда не сработает, что даст возможность агенту двигаться по более плавной и правильной траектории (см. рис. 1). Такая же проблема возникает, если агенту необходимо пройти вдоль нескольких препятствий, пример приведен на рисунке 2.

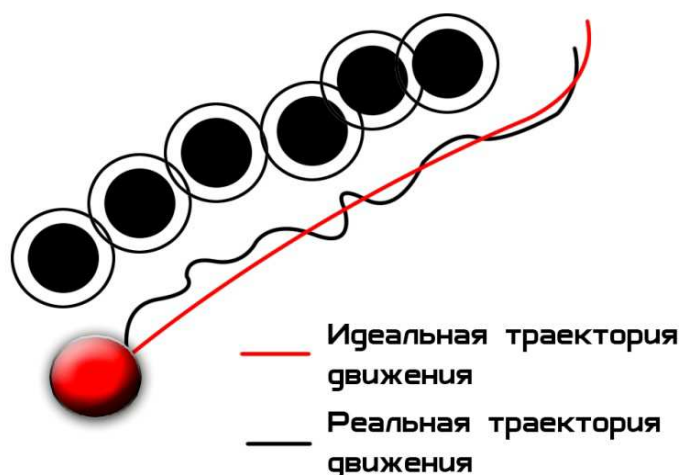


Рис. 2. Обход агентом несколько, подряд стоящих, препятствий

Для решения данной проблемы необходимо анализировать расстояние и угол направления к рядом стоящим препятствиям. Если расстояние достаточное и угол от препятствия довольно большой агент продолжает движение вдоль прямой, пока не покинет предел видимости всех препятствий, но если угол довольно маленький, агент начинает небольшой маневр, до достижения необходимого угла.

Следующая проблема, которая возникает при построении обхода препятствий это обход впереди стоящих 2-ух препятствий. Пример обхода 2-ух впереди стоящих препятствий показан на рисунке 3.

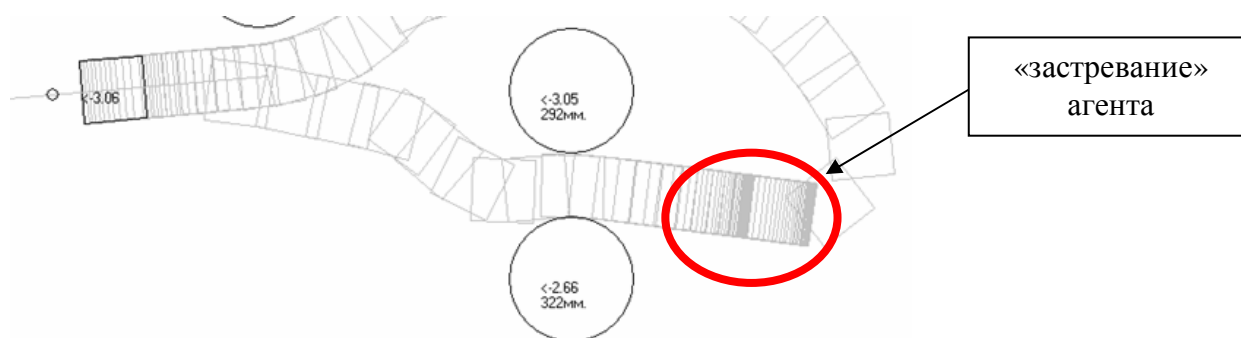


Рис. 3. Обход агентом 2-ух впереди стоящих препятствий

Из рисунка можно увидеть, что агент, пользуясь стандартными условиями, будет по началу пытаться от одного препятствия к другому, уменьшая расстояние до них. В момент, когда уже агент достиг препятствий, возникнет своего рода «застревание» агента между двумя препятствиями, через которые агент в течение длительного времени сможет протиснуться и поехать дальше. Чтобы решить данную проблему, необходимо знать расстояние между препятствиями, если его хватает, тогда агент спокойно проходит по прямой, в противном случае обходит одно из них. Так же чтобы достичь максимально плавный обход, необходимо поставить порог на вероятность срабатывания условий. Еще следует учесть тот момент, какому из препятствий отдавать приоритет, если вдруг расстояния между ними недостаточно, чтобы пройти без проблем, для этого просто в условиях отдаем больший приоритет в ту сторону обхода, которая необходима (лево или право, вверх или вниз).

Итак, 2 из 3 проблем рассмотрено, перейдем к 3-ей. Ее суть заключается в том, как будет вести себя агент по отношению к границе карты. При больших скоростях движения агента может возникнуть такая ситуация, что он выйдет за ее границу и вернется обратно. Пример данной ситуации показан на рисунке 4.

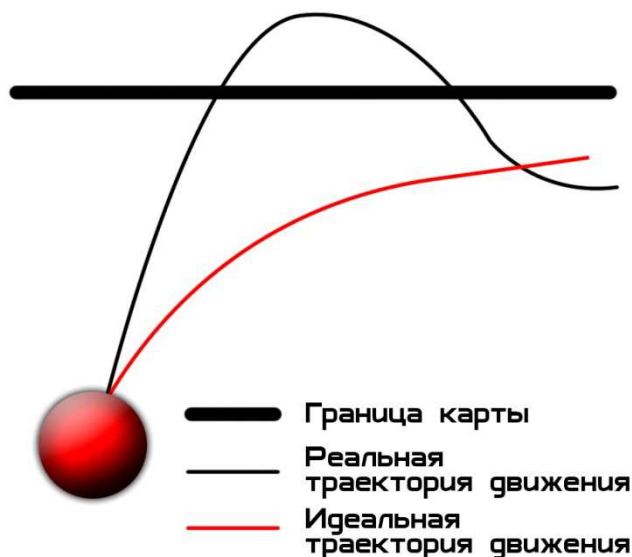


Рис. 4. Траектория движения агента вдоль границы карты

Для решения данной проблемы нам необходимо устанавливать расстояние между агентом и границей. Если агент движется с большой скоростью, а расстояние уже недостаточное для маневра, то он столкнется с границей и потеряет часть жизни. Если расстояния достаточно, то он сбросит скорость и начнет делать маневр и двигаться вдоль границы, при этом каждый раз высчитывается расстояние до границы и угол, на котором от агента она расположена.

На рисунке 5 показана работа программы по обходу препятствий агентами [2].

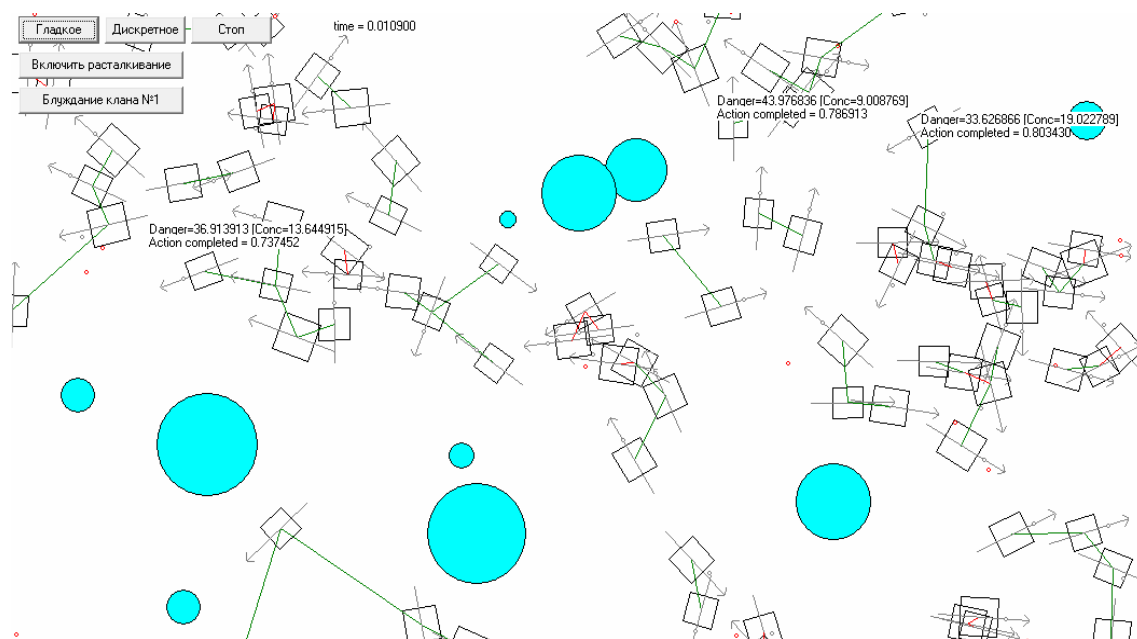


Рис. 5. Обход агентами препятствий

Таким образом, разрабатываемый нами проект будет решать все проблемы, которые были приведены выше, чтобы сделать ИИ в игре максимально простым в реализации и сложным в поведении.

ЛИТЕРАТУРА

1. ЭР – Нечеткая логика на практике [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/125614/>. – Дата доступа: 21.08.2015.
2. ЭР – Модель интеллектуальных агентов на нечеткой логике высшего типа [Электронный ресурс]. – Режим доступа: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/84682/02-Yershov.pdf?sequence=1>. – Дата доступа: 21.08.2015.
3. Шеври, Ф. Нечеткая логика / Ф. Шеври, Ф. Гели. – Вып. 31.

УДК 004.02

ОТОБРАЖЕНИЕ ОБЪЕКТОВ БАНКА НА КАРТЕ

С.В. РЕЗАНОВ

(Представлено: Е.Р. СУХАРЕВ)

Рассмотрен способ группировки объектов на картах Google Maps SDK. Дано подробное описание проблемы отображения объектов на карте и ее решение.

Перед реализацией научной работы проводилась оценка трудоемкости отдельных модулей приложения. Практически все приложение можно реализовать с помощью готовых средств. Однако самым трудоемким для разработки является группировка объектов на карте. Во время анализа не было найдено ни одного решения данной проблемы. Дело в том, что с появлением первой версии Apple iOS стандартным картографическим сервисом являлся «Google Maps». Многие сторонние разработчики реализовали библиотеки для группировки маркеров. Летом 2012 года с появлением iOS6 Apple отказались от всех сервисов Google, в том числе и от «Google Maps» [1]. Все приложения, которые использовали карты от Google, начали отображать карты от Apple. Таким образом, все ранее написанные библиотеки работали уже с «Apple Maps». В начале 2013 года Google выпустила первую версию «Google Map SDK» для iOS и активно продолжает развивать по сегодняшний день [2]. Однако уже реализованной возможности группировки маркеров нет. Так как карты от Apple не достаточно детализированы для России, а пользователями приложения будут в основном жители Московской области, то «Apple Maps» не подходит для реализации программы. Другие картографические сервисы работают не стабильно на новых версиях ОС. Поэтому альтернатив «Google Maps» нет. Принято решение реализовывать группировку маркеров самому.

Разработка алгоритма. Так как разработанный алгоритм не является простым и коротким в описании, то его сложно изобразить в виде блок-схемы, поэтому будет словесно описана основа алгоритма.